Claims

1. An alternation network for use with a content addressable memory for implementing a divide and conquer algorithm, the network comprising:

a plurality of alternation modules connected in series together, each module comprising:

a plurality of cascaded logic gates arranged to propagate a match parity signal via the gates along at least part of a matching result vector, the matching result vector being generated by execution of a matching instruction on the content addressable memory, and the logic gates being configured to change the parity of the match parity signal in accordance with the matching result vector; and

a vector output arranged to output a parity level vector of the propagated match parity signal present at the each gate of the plurality of logic gates;

a logic network for dividing the matching result vector into an odd match vector and an even match vector representing respectively odd and even numbered elements of the matching result vector, by use of the parity level vector; and

means for writing a selected one of the odd and even match vectors to the content addressable memory.

2. An alternation network of Claim 1, wherein the plurality of cascaded logic gates are arranged to implement a cascading exclusive-OR function on at least part of the matching result vector.

3. An alternation network of Claim 1 or 2, wherein each module further comprises a parity output for outputting a resultant parity of the plurality of cascaded logic gates, after the match parity signal has propagated through all of the logic gates of the module.

4. An alternation network of any preceding claim, further comprising generating means for generating a feed-forward term, the generating means being arranged to utilise the parity output of a neighbouring alternation module.

5. An alternation network of Claim 4, wherein the generating means is arranged to utilise a correction term, the correction term being generated from a non-neighbouring alternation module.

6. An alternation network of Claim 5, wherein the correction term represents the parity of a part of the matching result vector as determined by a section of the alternation network comprising a plurality of alternation modules.

7. An alternation network of any of Claims 4 to 6, wherein the vector output is arranged to be generated after a received feed-forward term has been applied to the module.

8. An alternation network of any of Claims 4 to 7, further comprising a plurality of interconnection pathways, means for determining a correction term from the parity of combined feed-forward terms, and pathways for feeding back the correction term to one or more alternation modules.

9. An alternation network of Claim 8, wherein the correction term is generated from a hierarchy of interconnection pathways.

10. An alternation network of any of Claims 4 to 9, further comprising a partition switch for each alternation module, the partition switch being arranged to disable propagation of the parity of the feed-forward term to any other alternation module.

11. An alternation network of any preceding claim, wherein the plurality of cascaded logic gates comprises a first cascade of logic gates arranged to calculate the parity of the matching result vector; and

a second cascade of logic gates for implementing an exclusive-OR function on the matching result vector using the received feed-forward term.

12. An alternation network of Claim 11, wherein the first cascade of logic gates is arranged to implement an exclusive-OR function on the matching result vector using a predetermined logic level input to a head of the plurality of cascaded logic gates.

13. An alternation network of any of Claims 1 to 10, wherein each module further comprises an injection module for receiving a starting logic level for the module and injecting the same into the plurality of cascaded logic gates.

14. An alternation network of Claim 13, wherein the injection module is arranged:

to assume an input logic level;

to inject this into the plurality of cascaded logic gates;

to compare the assumed input logic level with the received starting logic level and

to inject the starting logic level into the plurality of cascaded logic gates if there is a difference.

15. An alternation network of Claim 13 or 14, wherein the parity output is arranged to create the parity level vector from the injection of the assumed input logic level when the assumed input is correct and to create the parity level vector from the injection of the starting logic level when the assumed input is incorrect.

16. An alternation network of any preceding claim, wherein the logic network comprises an AND gate for generating an element of the even match vector from the parity match signal and the matching result vector.

17. An alternation network of any preceding claim, wherein the logic network comprises an inverter for inverting the propagated parity match signal

and an AND gate for generating an element of the odd match vector from the inverted propagated parity match signal and the matching result vector.

18. An alternation network of any preceding claim, wherein the logic network comprises means for storing the odd and even match vectors.

19. A combination of an alternation network as claimed in any preceding claim and a content addressable memory, the content addressable memory being arranged to receive the selected one of the odd and even match vectors for use as a matching result vector in a further iteration of the divide and conquer algorithm.

20. A combination according to Claim 19, further comprising a plurality of addition registers, each addition register being associated with an element of the content addressable memory.

21. A combination according to Claim 19 or 20, wherein the content addressable memory comprises a plurality of strings of array processors.

22. A communications network for use with a content addressable memory for moving data between remote locations, the network comprising:

dividing means for dividing a matching result vector generated by execution of a matching instruction on the content addressable memory, into an odd match vector and an even match vector representing respectively odd and even numbered elements of the matching result vector;

assigning means for assigning the odd and even match vectors as source and destination locations;

a plurality of cascaded logic gates arranged to propagate data via the gates along at least part of the matching result vector;

input means for inputting a bit of a stored data value associated with a source location into the plurality of cascaded logic gates;

applying means for applying the source and destination results to the plurality of cascaded logic gates to propagate the bit of stored data at the source location to the destination location; and

extracting means for extracting the input bit from the plurality of cascaded logic gates at the destination location.

23. A communications network of Claim 22, wherein the network is arranged to propagate a bit of the data stored at each source location to a corresponding destination location concurrently.

24. A communications network of Claim 22 or 23, wherein the plurality of cascaded logic gates is arranged to implement a cascading AND function.

25. A communications network of any of Claims 22 to 24, wherein the plurality of cascaded logic gates comprises a plurality of groups of sequentially connected logic gates.

26. A communications network of Claim 25, wherein each of the plurality of groups comprises an activation switch for activating or deactivating the respective group of sequentially connected logic gates.

27. A communications network of any of Claims 22 to 26, further comprising bypass means for connecting one group of sequentially connected logic gates to another non-neighbouring group of sequentially connected logic gates, thereby bypassing a neighbouring group of sequentially connected logic gates.

28. A communications network of Claim 27, wherein the bypass means is arranged to be activated when the neighbouring group of sequentially connected logic gates does not comprise any destination location.

29. A communications network of Claims 27 or 28, wherein the bypass means is arranged to be activated when the neighbouring group of sequentially connected logic gates is deactivated.

30. A communications network of any of Claims 27 to 29, wherein the bypass means is arranged to be activated when a plurality of neighbouring

groups of sequentially connected logic gates do not comprise any destination location.

31. A communications network of any of Claims 27 to 30, wherein the bypass means is arranged to be activated when a plurality of the neighbouring groups of sequentially connected logic gates are deactivated.

32. A combination of a communication network as described in any of Claims 22 to 31, a content addressable memory, and a plurality of addition registers.

33. A combination of Claim 32, wherein the combination is arranged to implement a remote addition algorithm, whereby data stored in the plurality of addition registers of selected source and destination locations are combined together using the communication network.

34. A combination of Claim 33, wherein the content addressable memory is arranged to receive the selected one of the odd and even match vectors for use as a matching result vector in a further iteration of the remote addition algorithm.

35. A combination of any of Claims 32 to 34, wherein the dividing means comprises an alternation network as claimed in any of Claims 1 to 18.

36. A combination according to any of Claims 32 to 35, further comprising a plurality of addition registers, each addition register being associated with an element of the content addressable memory.

37. A combination according to any of Claims 32 to 36, wherein the content address memory comprises a plurality of strings of array processors.